

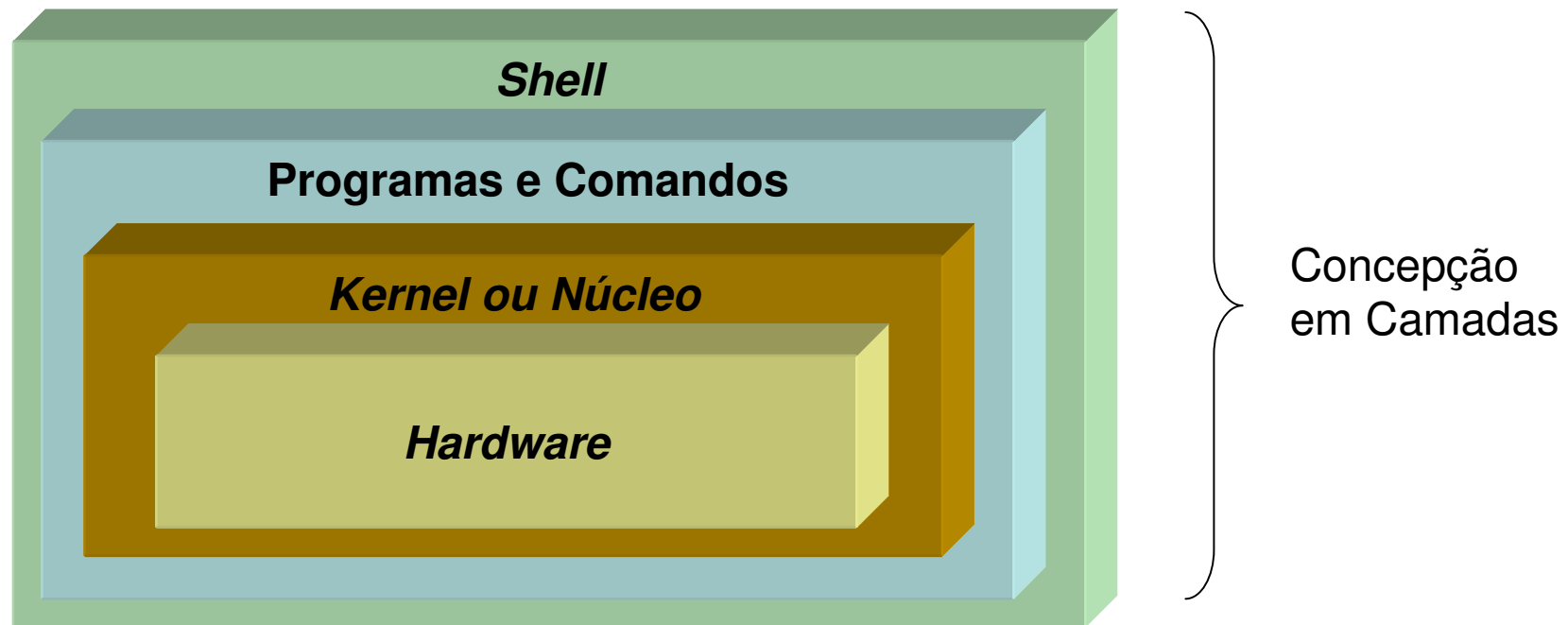
Programação Shell Linux

Conceitos Básicos

Objetivos

- Compreender a ordem de execução dos comandos pelo shell;
 - Conhecer os principais tipos de shell existentes.
-

Visão Geral do Unix



Visão Geral do Unix

■ Camada Shell:

- ❑ Responsável por interpretar os “desejos” do usuário;

```
$ cp /home/aluno/dados/* .
```

- ❑ O kernel executa o comando (**cp**), no entanto, **quem interpreta os metacaracteres é o shell, e não os comandos**

```
$ cd cur*  
$ pwd  
/meudir/curso
```

Tarefas do Shell (Ordem de Execução)

- Exame da linha de comandos:
 - 1º) O shell identifica os caracteres que têm significado especial para a interpretação da linha;
 - 2º) Verifica se a linha é um comando (ou programa) ou uma atribuição.

```
$ cp /home/aluno/dados/* . # comando
```

```
$ var=a #atribuição
```

Tarefas do Shell (Ordem de Execução)

- Avaliação de **Comandos**:

- Analisa a linha e identifica, **separadas por espaços em branco**:

- O nome do programa (qualquer comando é encarado como um programa);
 - Pesquisa a sua existência na ordem do caminho padrão (*path*);
 - Identifica suas opções/parâmetros;
 - Identifica seus redirecionamentos;
 - Identifica suas variáveis.
-

Tarefas do Shell (Ordem de Execução)

■ Atribuição:

- ❑ Identificada quando o *Shell* encontra um sinal de igualdade (=) separando dois campos sem espaços em branco;

<code>\$ var =a</code>	Procura o programa var por causa do branco
<code>\$ var=a b</code>	Espaço entre a e b, achou que b era comando
<code>\$ var='a b'</code>	Sem erro, o (') inibe a interpretação do seu interior
<code>\$ echo \$var</code> a b	Para o shell basta um branco como separador

Tarefas do Shell (Ordem de Execução)

- Resolução de direcionamentos:
 - ❑ O shell libera o kernel deste trabalho, preparando o ambiente de acordo com o tipo (*stdin*, *stdout*, *stderr*);
 - ❑ **Atenção:**
Se a saída de uma instrução for direcionada para um arquivo, este arquivo será criado vazio, antes deste comando ser executado.

```
$ cat arquivoTexto > arquivoTexto
```

Tarefas do Shell (Ordem de Execução)

■ Substituição de variáveis:

- ❑ Verifica se as variáveis, encontradas no escopo do comando, estão definidas e as substitui por seus valores.

```
$ echo $var
```

■ Substituição de meta caracteres:

- ❑ Substitui algum metacaractere (*, ? ou []) por possíveis valores.

```
$ cp arquivo?* file?*
```

Tarefas do Shell (Ordem de Execução)

- Passa a linha de comando para o Kernel:
 - ❑ Aí sim, completada a linha de comandos, o *Shell* monta a linha de comandos com todas as substituições feitas;
 - ❑ Chama o kernel;
 - ❑ O kernel cria um novo shell (*shell* filho) com um novo PID, que permanece inativo (*sleeping*) durante a execução do programa;
 - ❑ Encerra o processo (juntamente com o *shell* filho), recebendo novamente o controle.
-

Principais Shell's

- Interpretador de instruções (comandos), escrito em diferentes versões:
 - Bourne Shell
 - Padrão do Unix, escrita por Stephen Bourne (Bell Labs);
 - Também chamado de shell padrão (*standard shell*);
 - Sua representação é *sh*.
-

Principais Shell's

❑ Bourne-Again Shell

- Shell padrão do Linux;
- Compatível com o Bourne-Shell, e implementações do C-Shell e Korn-Shell;
- Sua representação é ***bash***.

❑ Korn Shell

- Desenvolvido por David Korn, da Bell Labs (AT&T);
 - *Upgrade* do Bourne-Shell;
 - Usado bastante pela comunidade Unix;
 - Sua representação é ***ksh***.
-

Principais Shell's

❑ C Shell:

- Desenvolvido por Bill Joy da Berkeley University;
 - Shell mais utilizado em ambientes Berkeley (BSD) e Xenix.
 - Estrutura bastante parecida com a linguagem C;
 - Não é portátil, pois programas desenvolvidos neste ambiente não funcionam em outros;
 - Sua representação para o Unix é ***cs*h**.
-